Санкт-Петербургский политехнический университет Петра Великого

Институт машиностроения, материалов и транспорта

Высшая школа автоматизации и робототехники

Отчёт

по лабораторной работе №2

Дисциплина: Техническое зрение

Тема: Открывающий фильтр

Студент гр. 3331506/70401 Водорезов Г.И.

Преподаватель Варлашин В. В.

« »\_\_\_\_\_\_\_\_2020 г.

Санкт-Петербург

2020

**Задание**

Пользуясь средствами языка С++ и библиотеки OpenCV, реализовать открывающий фильтр с границей типа border reflect для бинарного изображения с ядром приведенном на рисунке 1.

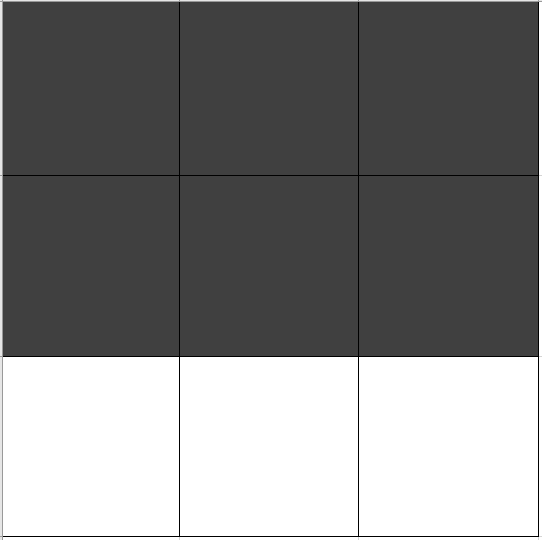


Рисунок 1 – Ядро фильтра

**Ход работы**

**Описание алгоритма**

Открывающий фильтр состоит в последовательном применении эрозии и дилатации с одинаковым структурным элементом. Его морфологический эффект заключается в удалении малых изолированных частей фигуры.

Применение эрозии сводится к проходу шаблоном по всему изображению и применению оператора поиска локального минимума к интенсивностям пикселей изображения, которые накрываются шаблоном.

Применение дилатации сводится к проходу шаблоном по всему изображению и применению оператора поиска локального максимума к интенсивностям пикселей изображения, которые накрываются шаблоном.

**Реализация алгоритма**

Для описания открывающего фильтра был создан класс *openingFilter*, изображенный на рисунке 2.

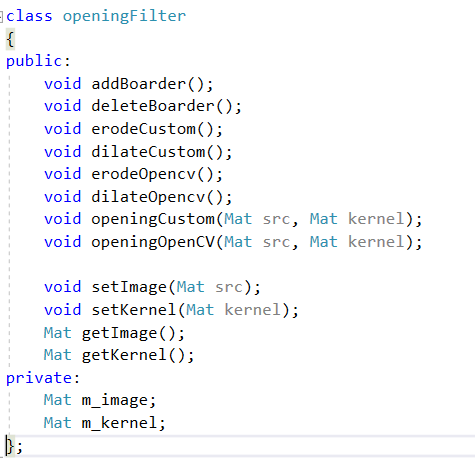


Рисунок 2 – Класс *openingFilter*

Данный класс реализован для бинарных изображений, перед тем как воспользоваться методом *openingCustom()* необходимо бинаризовать изображение. Код метода *openingCustom()* представлен ниже.

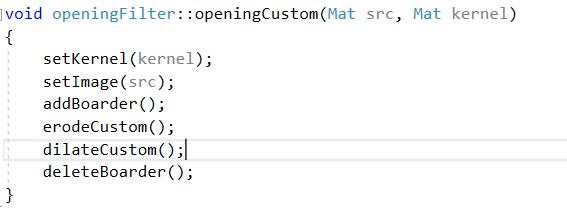


Рисунок 3 – Метод *openingCustom()*

Методы, используемые в *openingCustom()*, рассматриваются ниже.

Для расширения изображения реализован метод *addBoarder()*, код которого представлен на рисунке 4.

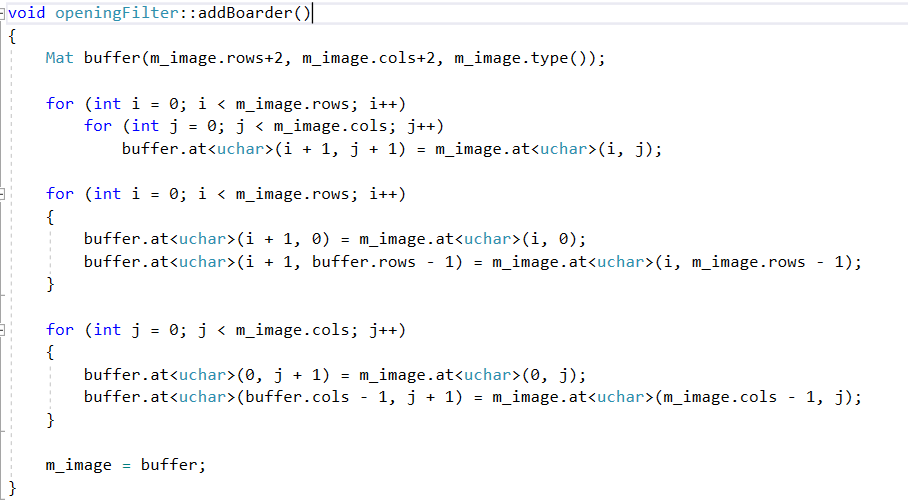


Рисунок 4 – Метод *addBoarder()*

Для осуществления эрозии был реализован метод *erodeCustom()*, код которой изображен на рисунке 5.

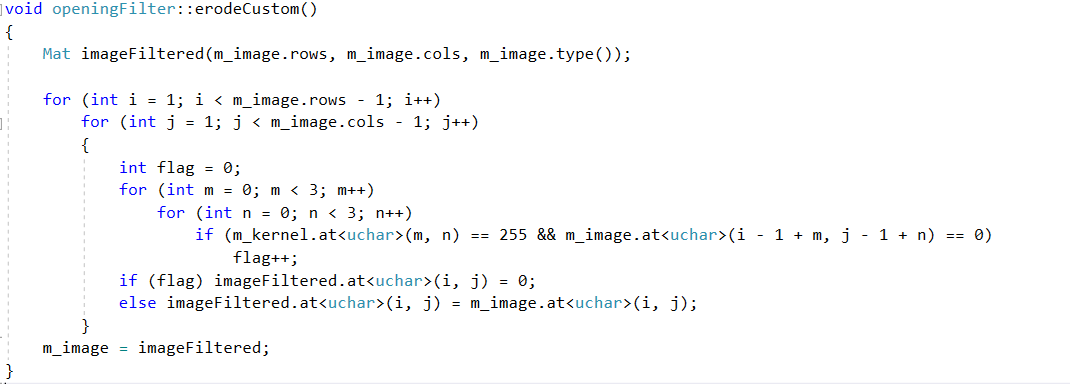


Рисунок 5 – Метод *erodeCustom()*

Для реализации дилатации изображения был создан метод *dilatCustom()*, код которого изображен на рисунке 6.

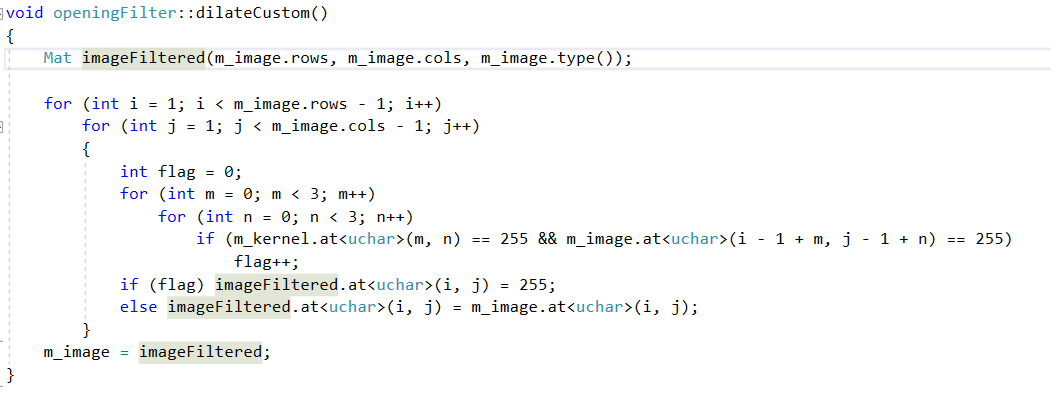


Рисунок 6 – Метод *dilateCustom()*

**Сравнение с методом из OpenCV**

Для обработки использовалось классическое изображение Лены Седерберг разрешением 800 на 800 пикселей (см. рисунок 7).



Рисунок 7 – Оригинальное изображение

Изображения, обработанные самописным фильтром и фильтром из OpenCV, приведены ниже на рисунках 8 и 9.



Рисунок 8 – Результат собственной реализации



Рисунок 9 – Результат реализации OpenCV

Для точного сравнения изображений реализован вспомогательная функция *comapreSimilarity()*. Функция выводит в консоль количество отличающихся пикселей. При сравнении результатов обработки данная функция вывела число 0, что соответствует полностью совпадающим изображениям.

Для сравнения времени выполнения обработки изображения использовался функцией *compareTime()*, которая выводит в консоль количество миллисекунд выполнения алгоритма в библиотеке OpenCV и собственной реализации.

Результаты определения времени открывающего фильтра изображения разрешением 800\*800 собственной реализации и реализацией из OpenCV представлены в таблице 1.

Таблица 1 – Сравнение времени выполнения

|  |  |  |
| --- | --- | --- |
| Конфигурация решения | Собственная реализация | OpenCV |
| Debug | 3265 мс | 6 мс |
| Release | 36 мс | 1 мс |

Различное время выполнения в различных конфигурациях связано с тем, что в конфигурации Release включена максимальная оптимизация, а в Debug оптимизация отключена полностью.

**Вывод**

В результате выполнения лабораторной работы был реализован открывающий фильтр. Результаты фильтра реализованным методом и методом из OpenCV полностью совпали, однако метод из OpenCV оптимальнее по времени выполнения.